Supplementary Material

Yuanqi Yao[1], Gang Wu[1], Kui Jiang[1], Siao Liu[2], Jian Kuai[1], Xianming Liu[1], and Junjun Jiang[⋆1]

[1] Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China
{yuanqiyao,kuaijian}@stu.hit.edu.cn
{gwu,jiangkui,csxm,jiangjunjun}@hit.edu.cn
[2] Academy for Engineering & Technology, Fudan University
saliu20@fudan.edu.com

1 Overview

In this supplementary material, we provide more details of SCAT as follows:

In Section 2, we provide detailed proof of the depth estimation error bound determined by the scaling coefficient k in the Scaling Depth Network (SDN), corresponding to Eq.11 in Section 4.2 of the main body.

In Section 3, we offer a exposition about the constant scaling coefficient k and the learnable scaling coefficient k, corresponding to the choice of coefficients discussed in Section 4.2 of the main body.

In Section 4, we present the experimental implementation details, parameter settings, and network architecture, aligning with Section 5 of the main text.

In Section 5, we present ablation studies and analyses on the size of the perturbations and the number of parameters within the adversarial noise generator.

In Section 6, we showcase qualitative experimental results in more corrupted scenarios within the complex out-of-distribution (OOD) dataset KITTI-C.

2 Proof of the Scaling Depth Network

Eq.11: Given the UNet-based depth network function defined:

$$DepthNet(x) = f_0(x),\ f_i(x) = b_{i+1} \circ [\kappa_{i+1} \cdot a_{i+1} \circ x + f_{i+1}(a_{i+1} \circ x)],$$

Assume $M_0 = \max\{||b_i \circ a_i||_2, 1 \leq i \leq N\}$ and f_N is L_0-Lipschitz continuous, where c_0 is a constant related to M_0 and L_0. Suppose I_t^ϵ is an input perturbed by $\epsilon = ||I_t^\epsilon - I_t||_2$, then we have:

$$||f_\theta(I_t^\epsilon) - f_\theta(I_t)||_2 \leq \epsilon \left[\sum_{i=1}^{N} \kappa_i M_0^i + c_0\right]$$

Proof: Let I_t^ϵ and I_t be the perturbed and vanilla inputs to the network, respectively. Consider the output difference caused by the perturbation:

$$||f_\theta(I_t^\epsilon) - f_\theta(I_t)||_2$$

⋆ Corresponding author

where f_θ represents the process of image reprojection through DepthNet D_t and PoseNet P_t. For simplicity, we temporarily disregard the error introduced by the reprojection process. Using the recursive definition of f_i, and expanding for f_θ, the difference is:

$$||f_0(I_t^\epsilon) - f_0(I_t)||_2,$$

using the recursive definition of f_i, we express the difference as:

$$||b_1 \circ [\kappa_1 \cdot a_1 \circ I_t^\epsilon + f_1(a_1 \circ I_t^\epsilon)] - b_1 \circ [\kappa_1 \cdot a_1 \circ I_t + f_1(a_1 \circ I_t)]||_2$$

Applying the triangle inequality and the norm properties:

$$\leq ||b_1 \circ a_1 \circ (I_t^\epsilon - I_t)||_2 + ||f_1(a_1 \circ I_t^\epsilon) - f_1(a_1 \circ I_t)||_2$$

Given $||b_i \circ a_i||_2 \leq M_0$, and considering f_N to be L_0-Lipschitz continuous, the above can be bounded by:

$$\begin{aligned} &\leq M_0 \cdot \epsilon + L_0 \cdot ||a_1 \circ (I_t^\epsilon - I_t)||_2 \\ &\leq M_0 \cdot \epsilon + L_0 \cdot M_0 \cdot \epsilon \\ &\leq (M_0 + L_0 \cdot M_0) \cdot \epsilon \end{aligned}$$

By iteratively applying this bound through all N layers of the network, we accumulate the factors of M_0 and κ_i, leading to the final inequality:

$$||f_\theta(I_t^\epsilon) - f_\theta(I_t)||_2 \leq \epsilon \left[\sum_{i=1}^{N} \kappa_i \cdot M_0^i + c_0\right],$$

which completes the proof.

3 Why Constant κ Works for SCAT

In this section, we propose the use of a constant scaling factor $\kappa = 0.7$ across all layers of the UNet-based depth network, a decision that, upon empirical evaluation, demonstrated superior performance compared to the use of layer-specific learnable scaling factors κ_i. This section delves into the theoretical analysis that might explain the observed efficacy of a constant scaling coefficient over learnable scaling coefficients $\{\kappa_i\}_{i=1}^N$.

3.1 Heuristic Constant Scaling Coefficients

To determine the most effective value for the scaling parameter κ within the UNet-based depth network, we considered a range of values within the interval $[0, 1]$. Through a heuristic binary search approach, we found that values greater than 0.5 were able to greatest retain the model's depth estimation performance on clean dataset while enhancing cross-domain generalizability. Subsequently, binary searching within the range $[0.5, 1]$ led us to adopt a constant $\kappa = 0.7$, which offered an optimal balance between performance on clean training dataset and generalizable to challenging scenarios.

3.2 Learnable Scaling Coefficients

Here, we introduce learnable scaling coefficients $\{\kappa_i\}_{i=1}^{N}$, each directly embedded within the respective decoding stage of our network. This integration of κ_i allows for a direct and adaptive adjustment of scaling factors during the network's training, fine-tuning the scaling to the distinctive feature map requirements at different layers.

3.3 Efficacy of Constant Scaling Coefficients

Simplification and Regularization. Employing a fixed scaling factor effectively simplifies the model's parameter space, acting as an implicit form of regularization. This simplification can help in preventing over-fitting, especially in scenarios with limited data availability, by restricting the model's complexity and thus enhancing its generalization capability.
Optimization Complexity. The introduction of learnable scaling factors κ_i increases the complexity of model tuning and the instability of the training process. Achieving optimal values for κ_i requires additional computational resources and time, potentially leading to sub-optimal local minima. In contrast, a fixed constant scaling factor streamlines the training process, possibly facilitating a more stable convergence to a global optimum.
Generalization and Universality. Empirical evidence suggests that a well-chosen constant scaling factor is sufficiently generalizable across different tasks and datasets, offering a simple yet effective solution. This approach reduces the necessity for over-tuning to specific tasks or datasets, thus increasing the model's versatility and adaptability.
In conclusion, while layer-specific learnable scaling factors introduce flexibility, our findings advocate for the simplicity, regularization benefits, and reduced training complexity offered by a fixed scaling factor, which collectively contribute to enhanced the generalization capability and stability of the UNet architecture.

4 Implementation Details

4.1 Experimental setup

To verify the generalization capability of the self-supervised MDE model to out-of-distribution scenarios, we use the KITTI dataset split of Eigen et al. [2] as our only training dataset, which results in 39,810 monocular triplets for training and 4,424 for validation. Following [3, 5–8], We use the same intrinsics for all images, setting the principal point of the camera to the image center and the focal length to the average of all the focal lengths in KITTI. During KITTI eigen split evaluation, we cap depth to $80m$ per standard practice. Moreover, We train the model starting with pretrained ImageNet weights [1] on RTX 3090 GPU. We use the Adam optimizer [4] for 30 epochs, with an input size of 640×192 and set a starting learning rate of 10^{-4}. Progressively reducing the learning rate using multi-step learning rate decay at epoch $[10, 20]$ by 0.1. We set the hyper-parameters ω, β and λ to 0.01, 0.01 and 0.001, respectively. The adversarial noise generator was trained with the Adam optimizer with a learning rate of 0.0001. We set ϵ_m to control the size of the perturbation to 135.0.

4.2 Architectures of Adversarial Noise Generator

Our adversarial noise generator g_ϕ consists of four layers. The first three layers are convolutional layers each with 20 filters of size 1x1, followed by ReLU activation function. The fourth and final layer is a convolutional layer with a channel depth of C, using a 1x1 filter. This setup maintains the spatial dimensions of the input throughout the layers, given the use of a stride of one and no padding, allowing for the generation of noise that is spatially uncorrelated.

5 Ablation Study

5.1 Effect of Parameter Counts within Adv Generator

In this section, we explore how varying the number of parameters in the adversarial noise generator affects the performance of self-supervised monocular depth estimation (MDE). The results of this investigation are detailed in Table 1. For clarity, we use subscripts to denote the depth of the noise generator used in each experiment. Except for this particular ablation study, all other experiments reported in this paper utilized a noise generator with a default setting of four layers. Our results suggest that while the depth of the adversarial noise generator is a tunable hyper-parameter, it does not significantly impact the performance outcomes for self-supervised MDE. As shown in Table 1, the most shallow adversarial noise generator, which consists of only one layer and 12 trainable parameters, leads to a decrease in mCE by approximately 2% compared to its deeper alternatives.

Table 1: Qualitative Results for different parameter counts within Adversarial noise generator. We compare the results obtained by 4 layers adversarial noise generator with its counterparts of different depths. Note that a depth of 4 layers was used in all experiments in this paper apart from this ablation study.

Model	Parameters	mCE(%)↓	mRR(%)↑	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
MonoDepth2 [3]	-	101.04	84.08	0.248	1.764	6.852	0.291	0.698	0.874	0.944
+ g_ϕ for 1 layer	12	88.24	88.67	0.172	1.583	6.231	0.273	0.755	0.891	0.949
+ g_ϕ for 2 layers	143	87.01	89.93	0.169	1.581	6.179	0.271	0.760	0.892	0.950
+ g_ϕ for 3 layers	563	86.73	90.05	0.168	1.579	6.162	0.271	0.761	0.893	0.951
+ g_ϕ for 4 layers	983	**86.32**	**90.13**	**0.165**	**1.573**	**6.157**	**0.269**	**0.762**	**0.893**	**0.951**
+ g_ϕ for 5 layers	1403	86.41	89.97	0.169	1.587	6.164	0.270	0.760	0.892	0.951

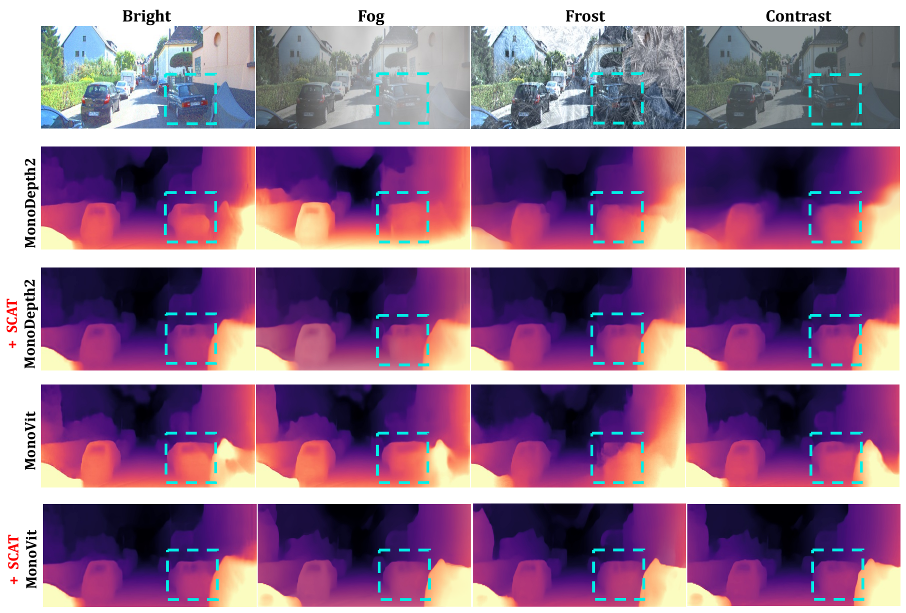

Fig. 1: Qualitative Results for Weather and Lighting KITTI-C.

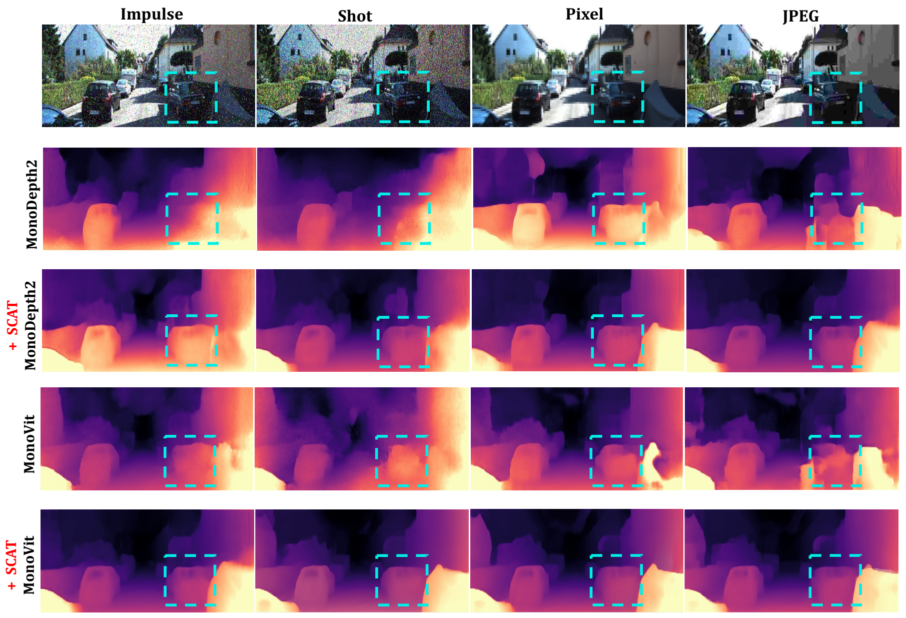

Fig. 3: Qualitative Results for Data and Processing KITTI-C.

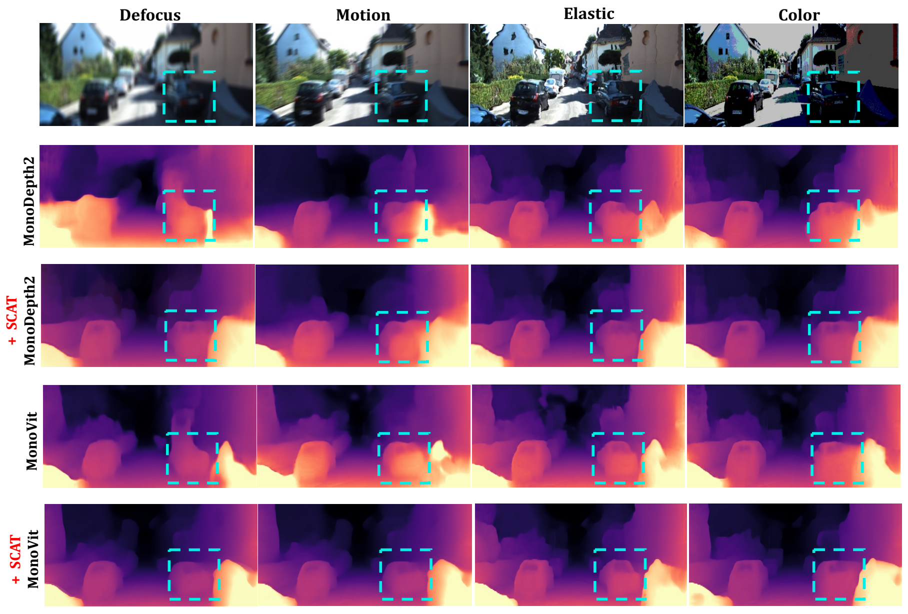

Fig. 2: Qualitative Results for Sensor and Movement KITTI-C .

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
2. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2650–2658 (2015)
3. Godard, C., Aodha, O.M., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth prediction. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3828–3838 (2019)
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
5. Lyu, X., Liu, L., Wang, M., Kong, X., Liu, L., Liu, Y., Chen, X., Yuan, Y.: Hr-depth: High resolution self-supervised monocular depth estimation. In: AAAI Conference on Artificial Intelligence (AAAI). pp. 2294–2301 (2021)
6. Yan, J., Zhao, H., Bu, P., Jin, Y.: Channel-wise attention-based network for self-supervised monocular depth estimation. In: IEEE International Conference on 3D Vision (3DV). pp. 464–473 (2021)
7. Zhao, C., Zhang, Y., Poggi, M., Tosi, F., Guo, X., Zhu, Z., Huang, G., Tang, Y., Mattoccia, S.: Monovit: Self-supervised monocular depth estimation with a vision transformer. In: IEEE International Conference on 3D Vision (3DV) (2022)
8. Zhou, H., Greenwood, D., Taylor, S.: Self-supervised monocular depth estimation with internal feature fusion. In: British Machine Vision Conference (BMVC) (2021)